GioMhail  
by CoolGle (Misha Kotlik & Gio Topa)

Basic protocols:

- Lesson on email structure, basic protocol info, SMTP commands and function, and using Java networking to send emails. (Read pages 1 & 2)

<http://www.quepublishing.com/articles/article.aspx?p=26672>

- SMTP Official Protocol Document (not anything new, but useful to have around)  
<https://www.ietf.org/rfc/rfc0821.txt>

- Working with Java Sockets (ignore the server part)  
<http://www.tutorialspoint.com/java/java_networking.htm>

- I/O, Files, and streams in Java  
<http://www.tutorialspoint.com/java/java_files_io.htm>

- Another example of using Sockets and SMTP in Java  
<http://www.java2s.com/Code/Java/Network-Protocol/SendingMailUsingSockets.htm>

- IMAP description on Wikipedia  
<https://en.wikipedia.org/wiki/Internet_Message_Access_Protocol>

- Latest official IMAP protocol (4rev1)  
<http://tools.ietf.org/html/rfc3501>

- POP3 Protocol:  
<https://www.ietf.org/rfc/rfc1939.txt>

More SMTP:

<http://www.samlogic.net/articles/smtp-commands-reference.htm>

More POP3:  
<http://www.electrictoolbox.com/article/networking/pop3-commands/>

- Do NOT use LAST command  
<http://www.dovecot.org/list/dovecot/2004-June/073667.html>

(To check for new emails, we would have to store list of email UIDs on client. Since we’re avoiding storage for now, we cannot implement an “unread” function.)  
- Extra commands: CAPA (capability) - returns list of capabilities including TOP, UIDL, and SASL types; AUTH - allows SASL auth with LOGIN or PLAIN

POP3 Security:  
<https://tools.ietf.org/html/rfc1734>

<https://tools.ietf.org/html/rfc5034>

<https://tools.ietf.org/html/rfc2449#section-6.3>

- Note, it’s a good idea to use auth plain if we can, the correspondence is shorter.

Mime:

- MIME on Wikipedia  
<https://en.wikipedia.org/wiki/MIME>

^ Follow links here for protocols

- Java mimetypes map might be useful

<https://docs.oracle.com/javase/7/docs/api/javax/activation/MimetypesFileTypeMap.html>

TSL/SSL:

<https://en.wikipedia.org/wiki/Transport_Layer_Security>

<http://info.ssl.com/article.aspx?id=10241>

<https://www.digicert.com/ssl.htm>

<https://www.symantec.com/page.jsp?id=how-ssl-works>

<https://docs.oracle.com/javase/7/docs/api/javax/net/ssl/SSLSocket.html>

<https://docs.oracle.com/cd/E11035_01/wls100/security/SSL_client.html>  
^This is probably over complicated, but explains some usage.

X<http://www.javacodegeeks.com/2013/06/java-security-tutorial-step-by-step-guide-to-create-ssl-connection-and-certificates.html>

<https://www.fastmail.com/help/technical/ssltlsstarttls.html>

<http://blog.trifork.com/2009/11/10/securing-connections-with-tls/>

<http://juliusdavies.ca/not-yet-commons-ssl-0.3.9/ssl.html>

><http://stackoverflow.com/questions/13135230/difference-between-ssl-and-tls-and-their-usage-in-java>

<http://stackoverflow.com/questions/8873855/how-to-implement-ssl-tls-secure-communication-through-normal-java-sockets>

^Explains how to upgrade connections following STARTTLS(I’m not sure if they’re handshaken?, need to test)

<http://gerardnico.com/wiki/smtp_telnet>

^Useful info about TSL & Telnet (Basically telnet doesn’t support TSL)

Use google from simple/test SMTP, POP3, and IMAP servers

Maildir (for local email storage)  
<https://en.wikipedia.org/wiki/Maildir>

* We would have to create our own system(classes) for working with maildir format.
* Folder full of files, and a directory file in folder to map file names to message ids/titles

Example APIs:

<https://github.com/ktheory/maildir>

<http://javamaildir.sourceforge.net/>

Gmail Specific/Examples:

<https://developers.google.com/gmail/oauth_overview?hl=en>

<https://developers.google.com/identity/protocols/OAuth2>

SASL & OAuth2:

<http://tools.ietf.org/html/rfc6749>

<http://tools.ietf.org/html/rfc4422>

Answer to SMTP issues

<http://serverfault.com/questions/280309/mail-server-on-port-25-only-response-with-500-command-not-recognized>

<https://commons.apache.org/proper/commons-net/examples/mail/SMTPMail.java>

WOOHOOO!

<http://stackoverflow.com/questions/7776369/how-to-determine-the-the-end-of-a-smtp-and-pop-response>

Existing CLI Email Clients:

<https://opensource.com/life/15/8/top-4-open-source-command-line-email-clients>

Mutt- good but complicated: <http://www.mutt.org/>

<https://www.google.com/search?q=mutt+client&source=lnms&tbm=isch&sa=X&ved=0ahUKEwifqq7P_afKAhWGtBoKHRFeCfMQ_AUIBygB&biw=1680&bih=925#imgrc=1xSXlG7_ioKl_M%3A>

Alpine-simpler and more elegant: <https://www.washington.edu/alpine/>

<https://www.google.com/search?q=alpine+client&espv=2&biw=1680&bih=925&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjdhfjK_6fKAhXBPRoKHTEeCvQQ_AUIBigB>

Sup & Notmuch aren’t as good. Alpine is the best.

Java CLI Frameworks:  
<http://doduck.com/command-line-interface-in-java/>

<https://commons.apache.org/proper/commons-cli/usage.html>

<https://github.com/vladimirvivien/clamshell-cli>

Best TUI/CLI UI Frameworks:  
<http://www.pitman.co.za/projects/charva/index.html> - Charva

<https://github.com/mabe02/lanterna> - Lanterna

CLI Parsing:  
<http://jline.sourceforge.net/> - JLine  
<https://commons.apache.org/proper/commons-cli/index.html> - Commons CLI

Java 7 Base64 Encoding:

<http://stackoverflow.com/questions/14413169/which-java-library-provides-base64-encoding-decoding>